(2)

AFHRL-TP-90-13

AIR FORCE

HUMAN RESOURCES

AD-A222 333

INTEGRATED MAINTENANCE INFORMATION SYSTEM (IMIS) DIAGNOSTIC MODULE, Version 4.0

Garth Cooke
Theodore Myers
Johnnie Jernigan
Nicola Maiorana

Systems Exploration, Incorporated
5200 Springfield Pike, Suite 312
Dayton, Ohio 45431

Dwayne Mason, Captain, USAF
Randy Link, Captain, USAF

LOGISTICS AND HUMAN FACTORS DIVISION
Wright-Patterson Air Force Base, Ohio 45433-6503

May 1990
Final Technical Paper for Period December 1988 – February 1990

Approved for public release; distribution is unlimited.

LABORATORY

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5601

90 05 29 018

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

JANET E. MURPHY, 2Lt, USAF
Contract Monitor

BERTRAM W. CREAM, Technical Director
Logistics and Human Factors Division

JAMES C. CLARK, Colonel, USAF
Chief, Logistics and Human Factors Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 1990

3. REPORT TYPE AND DATES COVERED
Interim - December 1988 to February 1990

4. TITLE AND SUBTITLE
Integrated Maintenance Information System (IMIS)
Diagnostic Module, Version 4.0

5. FUNDING NUMBERS
C - F33615-88-C-0004
PE - 62205F
PR - 1710
TA - 00
WU - 40

6. AUTHOR(S)
Garth Cooke
Nicola Maiorana
Theodore Myers
Dwayne Mason
Johnnie Jernigan
Randy Link

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Systems Exploration, Incorporated
5200 Springfield Pike, Suite 312
Dayton, Ohio 45431

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Logistics and Human Factors Division
Air Force Human Resources Laboratory
Wright-Patterson Air Force Base, Ohio 45433-6503

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFHRL-TP-90-13

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Air Force Human Resources Laboratory/Combat Logistics Branch (AFHRL/LRC) is engaged in the research and development (R&D) of an Integrated Maintenance Information System (IMIS). This system will be capable of accessing and integrating information from numerous Air Force data bases to provide technical support to the maintenance technician. This support will be provided by a portable computer maintenance aid which will provide instructions for accomplishing maintenance tasks. A diagnostic module will be contained in the portable computer software to help the technician in performing complex diagnostic tasks.

This paper describes the algorithms and decision logic employed in the IMIS module. The paper also describes the implementation of the diagnostic module in the portable computer used in a successful demonstration of integrated maintenance activities on the F-16 fire control radar system. (KR)

14. SUBJECT TERMS
criticality, multiple outcome test, symptom, dominant action, plausible set, diagnostics, exculpated fault, rectification

15. NUMBER OF PAGES
34

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFHRL Technical Paper 90-13 May 1990

INTEGRATED MAINTENANCE INFORMATION SYSTEM (IMIS) DIAGNOSTIC MODULE, Version 4.0

Garth Cooke
Theodore Myers
Johnnie Jernigan
Nicola Maiorana

Systems Exploration, Incorporated
5200 Springfield Pike, Suite 312
Dayton, Ohio 45431

Dwayne Mason, Captain, USAF
Randy Link, Captain, USAF

LOGISTICS AND HUMAN FACTORS DIVISION
Wright-Patterson Air Force Base, Ohio 45433-6503

Reviewed by

Robert C. Johnson, Chief
Combat Logistics Branch

Submitted for publication by

Bertram W. Cream, Technical Director
Logistics and Human Factors Division

This publication is primarily a working paper. It is published solely to document work performed.

SUMMARY

The Air Force Human Resources Laboratory/Combat Logistics Branch (AFHRL/LRC) is engaged in the research and development (R&D) of an Integrated Maintenance Information System (IMIS). This system will be capable of accessing and integrating information from numerous Air Force data bases to provide technical support to the maintenance technician. This support will be provided by a portable computer maintenance aid which provides instructions for accomplishing maintenance tasks. A diagnostic module will be contained in the portable computer software to help the technician in performing complex diagnostic tasks.

R&D in diagnostics has led to an IMIS diagnostic module which provides a wide range of capabilities that assist the technician in selecting an efficient sequence of maintenance tasks. These tasks lead to rapid and effective repair of failed components. The module was designed to work efficiently in an "on-equipment" maintenance environment. The technician's job in this environment uses the module to isolate problems to a replaceable component level rather than to the lowest possible level at which a failure might occur. However, the module is equally effective at the lower levels with appropriate adjustments to the data base.

The IMIS diagnostic module uses algorithms to identify the test and repair activity sequence most likely to result in a repaired system in the minimum amount of time. The algorithms calculate the likelihood of component failures and task accomplishment times in order to recommend the next sequenced action. The module determines these dynamically at each stage of the diagnostic session rather than exhaustively precalculating them to establish a fixed-sequence decision tree. Finally, the algorithms provide the technicians with lists of available actions which might prove effective in repairing the system. The lists are rank-ordered by calculated probability of success. The highest probability action is recommended; however, the technician is free to choose among the available alternatives. Once the technician completes an action, the next recommended action is then calculated based upon the results of the previous action.

This paper describes the algorithms and decision logic employed in the IMIS diagnostic module. This paper also describes the implementation of the diagnostic module in a portable computer used in a successful demonstration of integrated maintenance activities on the F-16 fire control radar system.

PREFACE

This paper documents the IMIS diagnostic module developed for the Air Force Human Resources Laboratory (AFHRL), Logistics and Human Factors Division, under the terms of contract F33615-85-C-0010 (Task Orders 0010-01, 0010-03, and 0010-09).

Captain Dwayne Mason and Captain Randy Link, AFHRL/LRC, were extremely helpful in the Research and Development (R&D) efforts to define and implement models and algorithms described in this technical paper.

The Dayton regional office of Systems Exploration, Inc. (SEI) performed the diagnostic research and computer model development. Principal investigators were Garth Cooke, Johnnie Jernigan, Michael Huntington, Nicola Maiorana, and Theodore Myers. They were assisted by Ronald Dierker and Colleen Gumienny. Human interface and portable computer development were accomplished by Captain Randy Link, Captain Dean Orrell, and Captain Gail McCarty of AFHRL/LRC and personnel from Systems Research Laboratories (SRL) including Jerry Brainard, Jane Slayback, and John Miles.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

LIST OF FIGURES

I. INTRODUCTION

Objective

This paper describes the current Integrated Maintenance Information System (IMIS) diagnostic module. The Air Force Human Resources Laboratory (AFHRL) is engaged in a long-term program to improve the effectiveness of information presentation in the maintenance environment. Research and development (R&D) in diagnostic aiding led to recognition, development, and coding of special features into an IMIS diagnostic module. This paper describes these features, the operation logic, and the implementation of these features.

Background

The IMIS diagnostic module helps the maintenance technician isolate and repair faulty aircraft components. The IMIS diagnostic module's key features are designed to minimize repair time rather than failure isolation time. This philosophy takes advantage of the instances when a rectification action would probably repair a problem faster than isolating the problem with tests and then repairing the problem. The IMIS diagnostic module has special subroutines that perform symptom/component matching, taking into account component histories, probabilistic data, logistic constraints, and operational constraints.

II. IMIS DIAGNOSTIC MODULE THEORY

We compared three baseline modeling techniques in developing the IMIS diagnostic aiding strategy. This comparison considered system history and design knowledge. The modeling techniques consist of fault modeling versus component modeling, fault isolation versus fault rectification, and information gained versus cost expended. Although each technique was evaluated as an independent approach, findings proved that combining beneficial attributes of related techniques was more effective. The following discussions provide descriptions of the comparisons and the combined attributes selected to develop effective diagnostic modeling techniques. The results accurately set up and attack the problem at hand, maximize the information gained, and minimize the cost expended.

Fault Modeling Versus Component Modeling

A component modeling technique maps each test result, fault code, or symptom to a plausible set of components. Rectification actions are then considered as a maintenance technician's action upon a single component. In flightline aircraft maintenance, problem rectification is frequently limited to "box swapping" or swapping of Line Replacement Units (LRUs). As a result, repair of broken or malfunctioning components as the goal of the diagnostic exercise is an accurate model. In addition, if the end item (the LRU as component) is disassembled to the Shop Replacement Unit (SRU) level, then modeling to the SRU as the component of interest would be an accurate model of a lower level of maintenance. A model based on this technique quickly becomes intractable due to large numbers of special cases and seemingly complicating information.

Fault modeling is an improvement over component modeling. Since most components are assemblies of lower level parts, failures of different parts in the component may have different effects. Any of these effects may constitute a malfunction of the component. Hence, when one defines a fault as the manifestation at the component level of a subcomponent's failure, then a component can be said to contain one or more potential faults. All of these faults can be readily defined through engineering analysis. The advantage of this technique is that faults are discrete, observable, or measurable while failures may be hidden; and the fault identifications are more descriptive than "malfunctioning component."

The objective of the diagnostic effort then is to isolate a fault rather than to isolate a faulty component. This fault modeling scheme greatly improves program effectiveness and tractability. The fault modeling scheme also provides significant amounts of failure data that may prove very valuable in subsequent maintenance activities at the SRU level. The problem with this technique is that the flightline maintenance technician swaps components rather than faults. This is a serious drawback because a model with a high level of fidelity to maintenance practice is essential to ready acceptance.

Therefore, we combined the two techniques. This solution was achieved by considering each component as "a bucket of potential faults." Each fault can be mapped to a rectification action. A rectification action may be the replacement of an LRU or some other maintenance action such as an adjustment, alignment, or a temporary repair. The set of rectification actions then maps to a single component upon which the rectification actions occur. After formulating and combining these two techniques, we produced a reachability matrix that mapped the diagnostic parameters of the system under investigation (see Figure 1).

In this matrix, the symptom (SO) implicates the faults (F0, F1, and F2) as potential causes of the observed problem. Each of the tests (T0, T1, etc.) is shown to span, or be affected by, one or more of the faults (shown as a 1). Test 3 is a Multiple Outcome Test (MOT) that has two discrete outcomes. It specifically measures for the presence of F1 and F2 in a single test procedure. Rectifications (R1 and R2) are maintenance actions that do not require removal or replacement of Component B. R3 is an action that requires removal and replacement of Component B.

					MOT					
	S0	**T0**	**T1**	**T2**	**T3**	**T4**	**R0**	**R1**	**R2**	**R3**
F0	1	1	1	0	0	F	1			
F1	1	1	0	0	1.1	C		1		1
F2	1	0	0	1	1.2	K			1	1
COMPONENT							**A**	**B**	**B**	**B**

F-Fault S-Symptom R-Rectification T-Test
MOT-Multiple Outcome Test FCK-Functional Check

Figure 1. Fault - Rectification - Component - Test Mapping.

Fault Isolation Versus Fault Rectification

The maintenance technician is frequently faced with the diagnostic problem of having two or more components in a system under investigation with no tests available to determine which of the components is faulty. The goal is to fix the system by replacing the components most likely to contain the failure and to minimize system downtime.

The initial assumption in developing the diagnostic module was that the technician would always attempt to isolate the faulty component (fault isolation) through available tests before attempting any repair action. Two factors led to this conclusion. First, fault isolation conserves supplies. Any attempt to repair prior to fault isolation can lead to the replacement of a component that is not faulty and needlessly depletes units from supply. Second, it conserves manpower by eliminating the effort required to remove and replace components that are not faulty.

This fault isolation strategy had to be reexamined. Given a particular symptom or set of symptoms, the set of possible faults may include a subset from one component that is so likely to have caused the symptom that an immediate rectification action is warranted. This sort of alternative may be particularly attractive when the system is badly needed for operational requirements. Under pressing time constraints, even when tests are available for fault isolation, analysis should provide a series of recommendations to repair a system in minimum time. However, if test times approach or become large compared to replacement times, the analysis might yield a swap first decision with a decreasing probability that the swap action will fix the fault. Such a situation could be very inefficient when there are no pressing time considerations or there are few spare components. To solve this problem, the Second Step probability of success was developed. This method provides an examination of what the maintenance technician could expect to find at the end of the second upcoming maintenance event in the diagnostic sequence.

Information Gained Versus Cost Expended

Initial development of the diagnostic algorithms and analyses focused on evaluating available options based on the information gained from the test results. This approach minimizes diagnostic steps in fault isolation. For example, several tests are frequently available in a diagnostics session to further the process. The task facing the maintenance technician is to select the most efficient test available. However, the information from a binary test can be a passing result and a failing result. A best test, therefore, is one which maximizes the information gained from whichever result occurs. We maximized the information gained by combining a split-half strategy with failure rates (FRs) of plausible faults.

I_j = The information value gained from performing test j
FR_i = The failure rate of the ith fault = 1/Mean Time Between Failure $(MTBF)_i$
$FR(1)$ = The failure rate of a plausible fault spanned by test j
$FR(0)$ = The failure rate of a plausible fault not spanned by test j
FR_{PS} = The failure rate of the plausible set = $\sum FR_i$

$$I_j = \frac{\sum_{i=1}^{PS} FR(1)}{FR_{PS}} \times \frac{\sum_{i=1}^{PS} FR(0)}{FR_{PS}} \qquad (1)$$

This strategy provided a means for selecting tests based on information gained but did not fully justify performing a time-consuming test. Other available tests may not provide as much information but may require a fraction of the time to complete. Certainly, time to accomplish a test should be considered a cost metric associated with that test. Excessive costs can accrue from an information gain strategy that maximizes the information gained but provides little insight into the cost of obtaining that information. This observation led to the development of the analyses that evaluate tests by calculating information gained per unit of time invested.

III. IMIS DIAGNOSTIC MODULE OPERATION AND ANALYSES

To develop the IMIS diagnostic module, we developed and employed algorithms that incorporate the above techniques while handling specific constraints inherent in aircraft data and maintenance applications. The diagnostic module operates in three major subdivisions: a) initialization, b) fault manipulation, and c) action ranking. During initialization, system descriptive data are loaded from a file system and specific constraint data are input through the computer keypad.

Faults are manipulated according to initial data entries and results of the technician's actions during the diagnostics session. Action ranking is performed recursively during the diagnostic session. It employs the analyses and calculations indicated by the current fault state. The current fault state is determined by the fault manipulation routines. This section explains the functionality and data processing for each of these activity subdivisions.

Logic Flow

Figure 2, Logic Flow, shows the sequencing of algorithms and analyses performed by the IMIS diagnostic module. In the initialization process, the IMIS diagnostic module accommodates both automatic and manual data input. Automatic data collection loads system specific data files from existing data bases and permits downloading of system health information from an aircraft data bus. The operator performs manual data entries such as symptoms, availability of parts and test equipment, critical states, and aircraft configuration. The diagnostic module then utilizes this information to evaluate fault combinations and to rank tests. Tests are then compared, by time analyses, to repair or replace activities, thus, obtaining the highest likelihood of fixing the problem in the least amount of time. Three lists of ranked tests and/or rectifications can be selected and presented to the maintenance technician: a) ranked tests, b) ranked rectifications, and c) interleaved tests/rectifications. Although, a "best" action is recommended, the technician is not prevented from choosing any of the listed options. When the technician selects a rectification or test, the presentation system displays technical order instructions for performing the selected activity. If the selected action is a test, the diagnostic module performs fault manipulations based on the test outcome and repeats the evaluation of available options. If the selected action is a rectification or maintenance action, the IMIS diagnostic module reinitializes the fault/symptom status utilizing changes in the system health information obtained from a functional check. This procedure continues until the fault is isolated, and the system is repaired.

Initialization

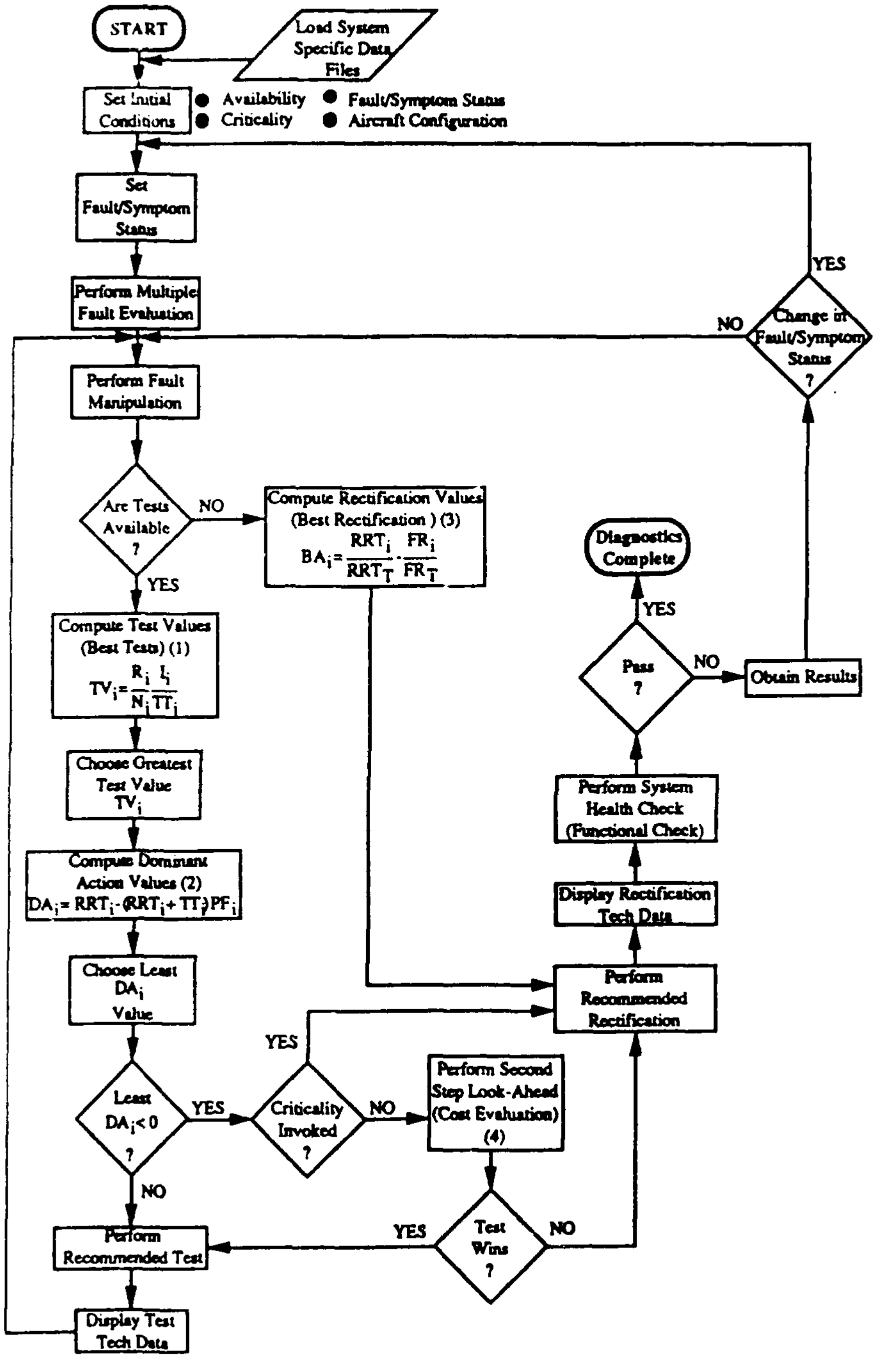
The initialization process provides the IMIS diagnostic module with pertinent information about the aircraft system under investigation, dictates the sequence of events to follow to solve the problem, and is essential for diagnostic analysis performance. The information supplied to the diagnostic module during initialization includes up-to-date system fault/symptom and action parameters, current aircraft system health information, availability of required parts and test equipment, criticality of repairing that specific aircraft system, and configuration of the aircraft system under investigation. The following paragraphs describe these inputs and their functionality within the IMIS diagnostic module.

Fault/Symptom Loading

To initialize the diagnostic module, system health information must be input from an outside source. System health information is entered as an observed malfunction of systems or machine-generated fault codes stemming from either automatic or operator-initiated Built-In Test (BIT). Possible symptoms and their associated potential faults are included in the system's data files. The observed symptom or fault code is the reason for starting a diagnostic session on a system.

Availability of Parts and Equipment

In many situations, part availability plays an important role in solving a maintenance problem. The IMIS diagnostic module takes this factor into account. At some point in the diagnostic session, the recommended action may be to remove and replace a component. Furthermore, such a recommendation may occur before all the plausible faults are isolated to a single component. In such a case, it would be unwise to have the diagnostic module present a



Notes:
(1) Section III. Page 10-13. Best Test/Multiple Outcome Test (MOT).
(2) Section III. Page 13. Dominant Action.
(3) Section III. Page 14. Best Rectification.
(4) Section III. Page 15. Second Step Look-Ahead.

Figure 2. Logic Flow.

recommended action which could not be met because the necessary components were not available through the base support system. Consequently, at the start of the diagnostic session, the technician is given an opportunity to annotate any parts which are known to be unavailable. The diagnostic module adjusts for unavailable parts and avoids making remove and replace recommendations for these parts until they become available or until other actions have isolated all remaining faults to the unavailable part(s).

The concept of availability can be extended to include the equipment necessary to complete the diagnostics and resulting maintenance actions. The test equipment availability feature of the IMIS diagnostic module takes into account the availability of test equipment and its direct effect on the ability to complete the recommended diagnostic tests. Consider a situation in which the IMIS diagnostic module has selected a test as the best option; however, the equipment to perform that test is presently inoperative or unavailable. This makes the test a less than optimal choice since it cannot be readily accomplished. The IMIS diagnostic module can consider test equipment availability when selecting the best option. This factor alleviates some frustration on the part of the technician faced with performing a test without the necessary equipment. Furthermore, this feature saves both time and money since the technician is warned against pursuing an action that cannot be performed. Once availability is set, the IMIS diagnostic module finds the tests and components that are affected, if any, and marks them for future reference. If an inappropriate action is selected, the IMIS diagnostic module displays the action as an invalid option.

Criticality

Criticality is a term used to designate some system functions essential for operational requirements. When referring to aircraft maintenance, the assignment of critical functions signifies that potential faults in those functions must be fixed or confirmed as good before the aircraft leaves for its next mission. As an example, assume that a weapon system has both air-to-air and air-to-ground capabilities. If the next scheduled mission is for an air-to-air combat capability, then air-to-air capabilities may be designated critical and air-to-ground capabilities noncritical.

During initialization, the technician has an opportunity to designate a function or group of functions critical. All potential faults contained in the components required to accomplish the critical function are then identified as critical. The diagnostic module then searches for plausible faults (implicated by symptoms) identified as critical. This group is designated the critical set of plausible faults. The critical set is then given special consideration in developing recommended actions as explained on page 16 under the section entitled "Criticality."

Aircraft Configuration

System configuration is an important consideration for any diagnostic aid because as configurations change, the set of valid plausible faults will also change. If the diagnostic aid does not consider changes in system configuration, it will provide incorrect or misleading results. For example, consider an F-16 which is configured completely with conventional weapons but has a nuclear Remote Interface Unit (RIU) installed in one of the pylons. On performing a system BIT, one of the symptoms that will appear on the Fire Control Navigation Panel display is lost communication with the nuclear RIU. However, this error message is normal for the F-16's all-conventional weapons configuration. Due to configuration irregularities, a symptom is present that is normal for the current configuration. If the symptom is considered a valid problem indication, incorrect diagnostic sequencing will inevitably occur.

In order to avoid such confusing circumstances, the IMIS diagnostic module is notified of the aircraft's configuration during the initialization sequence so that appropriate symptoms are ignored and appropriate faults eliminated from consideration prior to diagnostics.

Evaluation of Faults

The IMIS diagnostic module provides a multifaceted approach to the evaluation of potential faults, combining them into sets using the multiple fault evaluation, partitioning possible faults from unlikely faults, and attacking the suspected combinations that most likely are the cause of an inoperable or malfunctioning system. The idea of attacking more than one possible fault at a time is a new development in aircraft diagnostics and is incorporated in all the decision-making algorithms presented in this section.

Multiple Fault Evaluation

"Multiple faults" are two or more faults that occur simultaneously. Multiple faults can appear in a system in a variety of ways. The algorithms employed in the IMIS diagnostic module are designed to handle all types of multiple faults that might occur including:

1. Single Symptom. This type of multiple fault arises when a single symptom is identified; however, two or more failures might actually be causing that symptom.

2. Multiple Symptom. This is a more complex type of multiple fault. In this case, multiple symptoms are present which can be caused either by a single fault or by a combination of faults.

The IMIS diagnostic module attacks the problem of multiple potential faults by considering several factors: distribution of fault probabilities for the symptoms being considered, how the symptoms span the set of possible faults, the lower probability of independent events occurring simultaneously, and the influence of the time required to complete each possible action. Consider Figure 3, a multiple fault scenario including three symptoms and four faults:

	F1	F2	F3	F4
S1	$P(F_{11})$	$P(F_{21})$	$P(F_{31})$	0
S2	$P(F_{12})$	$P(F_{22})$	0	0
S3	$P(F_{13})$	0	0	$P(F_{43})$

Figure 3. Fault/Symptom Probability Matrix.

The probability that a specific fault caused a specific symptom is indicated by the probabilities in the array. For example, the probability that F1 caused S1 is $P(F_{11})$, the probability that F4 caused S3 is $P(F_{43})$, and the probability that F4 caused S2 is 0 (S2 is not caused by F4). This scenario can be extended to each symptom and fault combination. This model also shows how the symptoms span the set of faults. A symptom's spanned faults are all of the nonzero entries for the row indicated. The fault probabilities are obtained from system data files.

Since fault occurrences are considered independent events, if the probabilities of each event can be computed, then the probabilities of combinations of these independent events can also be computed. This conclusion follows from the fact that the probability of independent events occurring simultaneously is the product of the individual probabilities of each independent event. Therefore, fault combinations that could have caused the set of symptoms can be listed and the corresponding probabilities computed according to this rule.

Once the probability of each combination is computed, the combinations are then rank-ordered, the one having the highest probability being the most likely to have caused the set of

symptoms. Reduction of this initial plausible set of fault combinations is accomplished by removing redundant fault combinations from consideration. A redundant fault combination is a combination that is repeated in another combination.

1. Compute individual fault probabilities for n symptoms and m faults. $P(F_{ji})$ is the probability that fault F_j has occurred given symptom S_i has been detected.

$$P(F_j)_{j=1,m} = \frac{\sum_{i=1}^{n} P(F_{ji})}{n} \qquad (2)$$

2. List possible fault combinations.

3. Compute probability of the combinations of independent faults.

$$P(FC_i) = \prod P(F_j) \qquad (3)$$

where $P(F_j)$s are the independent fault probabilities calculated in (2) that combine to make the fault combination FC_i. (i = 1 to number of unique combinations.)

4. Rank in order of highest probability and remove redundant choices (remove redundant $P(FC_i)$s). This methodology examines the list of possible fault combinations and removes from consideration those combinations that are included in a more probable combination. For example, if the method selected F2F4 as the most probable combination, any other combinations including F2F4 will be removed from the set because, by operating on and exculpating F2F4, these combinations will become invalid (given the same set of symptoms). This considerably reduces the plausible set.

5. Compute the probability of fault combinations as a probability of the reduced plausible set allowing the sum of all $P(FC_i)$s in the reduced plausible set to equal one.

$$P(FC_i)_{PS} = \frac{P(FC_i)}{\sum_{i=1}^{nc} P(FC_i)} \qquad (4)$$

where $P(FC_i)_{PS}$ = probability of Fault Combination i of the reduced Plausible Set,
$P(FC_i)$ = probability of Fault Combination i as computed in (3), and
nc = total number of combinations in reduced Plausible Set.

Fault Partitioning through Fault Manipulation

In performing fault manipulation, faults are moved from set to set, presenting new fault combinations to attack, saving fault combinations removed from current consideration, and exculpating potential faults due to passed tests and functional checks. Figure 4, Fault Manipulation, provides a graphic view of fault movement during the diagnostic process.

Upon initialization, the diagnostic module performs a multiple fault evaluation and produces a plausible set of potential fault combinations. The plausible set of potential fault

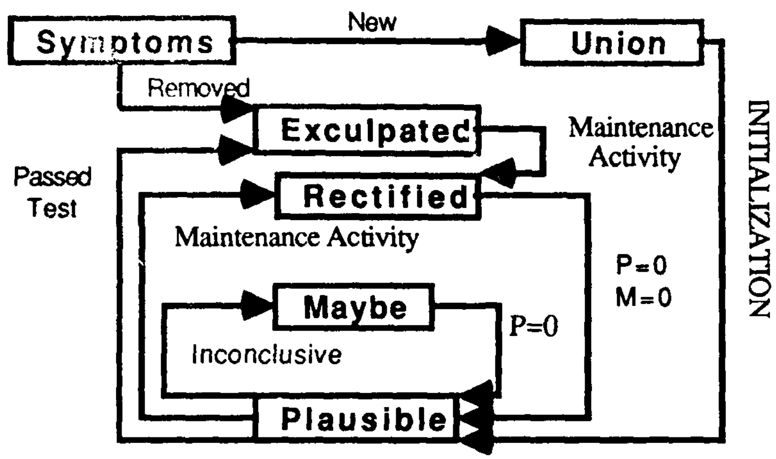


Figure 4. Fault Manipulation.

combinations is evaluated and presents a ranked list of options. Based on the option selected and its results, several fault manipulations can occur:

1. Completing a rectification results in spanned faults being placed into the rectified set. This manipulation groups all rectified faults and saves them. If the plausible set is exhausted and the system malfunction still exists, then the rectified set will be used for continued diagnostics. One cannot assume all replacement units from supply are good. Completion of the rectification also prompts a functional check for system health information. If a "pass" on the functional check is observed, all potential faults are exculpated and diagnostics end. Conversely, if a "fail" is observed, the functional check returns updated symptom information. If there are no changes in the symptom status, diagnostics continue with the current plausible set. If there are changes in the symptom status, the multiple fault evaluation is performed with the new symptom status and a new plausible set is produced.

2. Fault manipulation after a test depends on the test outcome. If a "pass" is observed, potential faults associated with that test are exonerated. They are placed into the exculpated set and eliminated from consideration. Faults in combination with the exculpated faults are placed in the maybe set, provided they are not part of another fault combination in the plausible set. If a "fail" is observed, one or all of the faults spanned by that test are known bad. These faults and their combinations remain in the plausible set for the next action ranking, and all other fault combinations are placed into the maybe set for future evaluation.

When the plausible set has been exhausted, faults are transferred from the maybe set into the plausible. This fault manipulation process repeats until diagnostics are successfully completed or until symptoms are returned and no faults are left in the maybe or plausible sets. When this situation occurs, the faults from the rectified set are placed in the plausible set and diagnostics continue.

Action Ranking

Upon performing the multiple fault evaluation, the diagnostic module uses plausible fault combinations to evaluate the diagnostic actions available for isolating and repairing the aircraft system. A split-half strategy is incorporated into the best test and MOT algorithms to obtain the most information gained per unit of invested time. The best test is then ranked against available

actions using the action ranking routines. Hence, the module determines the highest likelihood of fixing or isolating the problem in the least amount of time and cost. Included in action ranking are the dominant action, rectification, and second step look-ahead analysis.

Split-Half Strategy

The diagnostic module supports a split-half troubleshooting strategy. The initial symptoms' spanned set of potential faults determines the initial plausible set (that set in which at least one fault must exist). Each test's intersection with the plausible set is evaluated. The test that most nearly divides the initial set in half is selected as the next best test. This process is repeated until the plausible set has only one component or until no tests are available that reduce the size of the plausible set. In the latter case, a brute force method of exchanging components is adopted.

A split-half strategy will always isolate a fault in the fewest number of steps whenever test times and fault probabilities are equal. However, test times and fault probabilities are rarely equal; moreover, other constraints also have significant bearing upon the selection of an appropriate diagnostic strategy. Consequently, in developing the action ranking routines, the following additional options were implemented:

Best Test	Second Step Look-Ahead
MOT	Criticality
Dominant Action	Interleaving Actions
Best Rectification	

Best Test

The information gained from a binary test is reflected in both the pass result and the fail result. A best test is one which maximizes the information gained from whichever result occurs. Different tests frequently consume different amounts or kinds of resources. That is, there is a cost function associated with the choice of a best test. A commonly available metric about which cost can be allocated is the time to perform the test. Therefore, test time or task time has been used throughout this project as the basic cost metric. In this case, we have chosen to evaluate the best test for maximum information gained per unit of time. Consequently, the best test evaluation used in this program has been defined as the following:

$$BT = \max \frac{I_j}{T_j}\text{, where} \tag{5}$$

$$I_j = \frac{\sum_{i=1}^{PS} FR(1)}{FR(PS)} * \frac{\sum_{i=1}^{PS} FR(0)}{FR(PS)} \tag{6}$$

T_j = time to accomplish test j.

Selecting a best test for a multiple fault problem increases the complexity of the problem and requires a special technique. The IMIS diagnostic module has implemented the following technique which constructs a partitioned set of probable faults and ranks tests to select a best test under multiple fault conditions. The first step is to partition the set of faults into fault combinations that have a high probability of having caused the symptoms. This step is accomplished by rank-ordering the fault combinations in descending order and removing redundant combinations. This is

the first partition. The second is created by grouping tests that provide the same information about a fault combination. Consider the example in Figure 5. Figure 6 below depicts all combinations of tests that may be available to the technician. Tests associated with F_B will evaluate F_2 or F_4 independently or as a combined set (F_2F_4). Evaluation of either fault independently or as a combined fault will result in the same information because both F_2 and F_4 must be present for the three symptoms to have occurred. Therefore, the span of tests can be reduced to T_A, T_B, T_C as shown in Figure 7.

	S1	S2	S3	T1	T2	T3	T4	T5	T6	T7
F1	1	1	1	0	0	0	1	1	1	1
F2	1	1	0	0	1	1	0	0	1	1
F3	1	0	0	0	0	0	0	0	0	0
F4	0	0	1	1	0	1	0	1	0	1

Figure 5. Multiple Fault System Model (S1, S2, S3 all exist).

	F_A	F_B	
	F_1	F_2	F_4
T1	0	0	1
T2	0	1	0
T3	0	1	1
T4	1	0	0
T5	1	0	1
T6	1	1	0
T7	1	1	1

Figure 6. Sample Fault/Test Relationship Matrix.

T_A={T1,T2,T3}=T1=T2=T3
T_B={T4}
T_C={T5,T6,T7}=T5=T6=T7

	$P(F_1)$ F_A	$P(F_2)P(F_4)$ F_B	Test Weight
T_A	0	1	$P(F_1) \times P(F_2)P(F_4)$
T_B	1	0	$P(F_2)P(F_4) \times P(F_1)$
T_C	1	1	$0 \times [P(F_2)P(F_4) + P(F_1)] = 0$

Figure 7. Reduced Fault/Test Matrix.

This multiple fault system shows three tests spanning two faults. In this system, test groups T_A and T_B provide the best split of the faults. Since T_C completely spans both faults F_A (F1) and F_B (F2F4), it is guaranteed to fail. A pass on test group T_B will exculpate F_A, and a fail will implicate F_A. These test groups can then be inserted directly into the best test formula.

This technique eliminates tests guaranteed to fail, suspends from consideration tests with ambiguous outcomes, and operates upon fault sets generated by the multiple fault algorithms. The test group(s) with the highest score can then be selected as the best test(s). The best test(s) are then compared to available maintenance activities using the dominant action analysis. The results of this process are used in conjunction with the normal diagnostic path until the problem is solved.

Multiple Outcome Test (MOT)

A test which has MOTs creates special problems when trying to measure its worth against other available tests that also split a plausible set. This problem is because a test with multiple outcomes is not binary (a pass is not the complement of a fail). A purely binary test will result in the exoneration of all spanned faults in the event of a pass and the inclusion of all spanned faults in the event of a fail. Conversely, a test with multiple outcomes, an MOT, would exonerate all spanned faults in a pass condition but include only a restricted number of spanned faults in one of several possible fail conditions. Any one of several possible fail results can lead to isolating faults to a number much smaller than the test's spanned set. Therefore, an MOT is generally more powerful than a binary test which spans or splits the same set; hence, it is more valuable.

A perfect MOT is one in which each outcome isolates a single fault, and there are sufficient outcomes so that each fault spanned by the MOT can be isolated, as shown below:

	SPAN
	0 0 0 0 0
	1 0 0 0 0
OUTCOMES	0 1 0 0 0
	0 0 1 0 0
	0 0 0 1 0
	0 0 0 0 1

Conversely, a poorly designed MOT would neither isolate single faults nor contain sufficient outcomes, as follows:

	SPAN
	0 0 0 0 0
OUTCOMES	0 1 1 1 1
	1 1 1 1 0

From the above, it can readily be seen that the value of an MOT is related to the "sparseness" of dependencies coupled with the number of possible outcomes. Consequently, it was our intent to develop a relationship that takes advantage of these logically and aesthetically obvious relationships. As the best test from a set of binary tests can be determined from equation (5), the logical approach to accommodating MOTs was to operate directly upon the best test algorithm (repeated in equation (7)).

$$BT = \max \frac{1}{T} \times \frac{\sum_{i=1}^{PS} FR(1)}{FR(PS)} \times \frac{\sum_{i=1}^{PS} FR(0)}{FR(PS)} \tag{7}$$

Examination of the best/worst MOT displays showed that a scale factor that describes the "sparseness" of dependencies can be readily determined from the ratio where the zeros and ones are counted for all test outcomes.

$$R = \frac{\sum^{rows} 0\text{'s}}{\sum^{rows} 1\text{'s}} \qquad (8)$$

In addition to looking at the relative efficiency of the test through the R ratio, we must also evaluate the value of the individual test outcomes. This task can be accomplished using the same algorithm for best test and then computing the average for all test outcomes as shown below:

$$\bar{I} = \frac{1}{n} \sum_{j=1}^{n} \left[\frac{\sum_{i=1}^{PS} FR(1)}{FR(PS)} \times \frac{\sum_{i=1}^{PS} FR(0)}{FR(PS)} \right] \qquad (9)$$

where $\bar{I}$ = the average information gain from the MOT
n = number of outcomes.

The revised best test algorithm then becomes the following:

$$BT = \max \frac{R\bar{I}_j}{T_j} \qquad (10)$$

Having established this algorithm as an accurate measure of the value of a MOT, it was then necessary to look at how this algorithm affected the result from evaluating a binary test. If R and n are defined as one for binary tests, then the above equation reduces to the original best test algorithm. Hence, this solution to the MOT problem allows a single equation to be used to compute all best test values.

Dominant Action

Given a particular symptom or set of symptoms, the plausible set may contain a particular component that is so predominantly likely to be the cause that an immediate rectification action is warranted. This sort of alternative may be particularly attractive under certain criticality considerations. In these cases, resource conservation can become a secondary consideration.

In order to examine this change of philosophy, we needed to establish a mathematical relationship that measured rectification time against the choice of methods.

Let: RT = Rectification Time
TT = Test time
RRT = Removal and replacement time
PFn = Probability that fault n has occurred

Assume there are two strategies available. Strategy 1 is to perform an immediate replacement of the most likely cause of the faults in the plausible set without any attempt at fault

isolation; then, if necessary, perform a fault isolation test if one is available. Strategy 2 is to perform diagnostic testing to isolate the faulty component first. The choice between Strategy 1 and 2 can be made on a particular component based on the following dominant action equation.

$$\begin{aligned} \text{Delta RT} &= \text{RRT} - (\text{RRT} \times \text{PF}_n + \text{TT} \times \text{PF}_n) \\ &= \text{RRT} - (\text{RRT} + \text{TT}) \times \text{PF}_n \end{aligned} \tag{11}$$

This formula determines the difference between the time to rectify the component (RRT) and the time to perform the test (TT) first plus rectify (RRT) that component using the probability (PFn) that fault n occurred.

Therefore, if Delta RT < 0, Strategy 1 (swap first) is the best option, signifying that the probability of that component being faulty is so high that it should be replaced without testing and/or testing time is high compared to rectification time. If Delta RT > or = 0, Strategy 2 (test first) is the best option because the probability of that fault having caused the problem is not very likely and/or the rectification time is very high, and it is better to test before rectifying.

Under pressing time considerations, dominant action recommendations will generate a fixed component in minimum time. However, if test times approach or become large compared to replacement times, the equation yields a swap first decision with a smaller and smaller probability that the swap action will fix the fault. Such a situation could be very inefficient when there are no pressing time considerations or there are few spare components. To solve this problem, the Second Step probability of success was developed. It provides an examination of what the maintenance technician could expect to face at the end of the second upcoming maintenance event in the diagnostic sequence. This analysis is discussed in Section III, Action Ranking (Second Step Look-Ahead).

Best Rectification

If no tests are available, diagnostics must be completed by rectifications alone. The best rectification routine recommends the best rectification to perform first based on time to rectify and probability of failure. This routine is also performed to provide the maintenance technician with the human interface feature of showing a ranked list of the five best actions.

The best rectification analysis provides a strategy to minimize the total time to system rectification. The following variable definitions apply:

RRT_i = Remove and Replace time of the ith component considered by the plausible set of faults.
RRT_T = Sum of Removal and Replacement times for all components considered by the plausible set of faults.
FR_i = Failure rate of a given fault or component = 1/MTBF
FR_T = Failure Rate of the plausible set.

The results of the analysis are generalized to a multicomponent system if we recognize that individual comparisons provide only a relative ranking between components. Failure to achieve success on the first option requires recycling through the algorithm to determine the next best option. Therefore, the generalized form of the algorithm to rank the trade-off between failure rate and substitution time is as follows:

$$BR_i = \frac{RRT_i}{RRT_T} - \frac{FR_i}{FR_T} \tag{12}$$

for the ith component.

Results from this analysis will range from -1 to 1. The component with the lowest BR_i is the optimum candidate for the best rectification. All of the actions are ranked from lowest to highest for ranked rectification list.

Second Step Look-Ahead

The second step look-ahead analysis provides a diagnostic recommendation based on the cost difference between the dominant action and the best test by analyzing what the maintenance technician could expect to face at the end of the second upcoming maintenance event (next activity) in the diagnostic sequence. Upon completion of the dominant action analysis, the dominant action recommendation can take two routes depending on the state of system criticality. If criticality is invoked, the diagnostic module automatically recommends performing the dominant action because cost is not a factor if the system is to be repaired in minimum time. If the system is not deemed critical for the next mission, the second step look-ahead analysis is performed.

When second step look-ahead is chosen, two viable diagnostic activities, a best test and a dominant action, are available to continue diagnostics. The unit cost of the dominant action and the best test are calculated using the following formulas. The activity with the lowest cost is recommended as the next activity to be performed.

To correctly perform this analysis, one must realize that a test cannot fix a system (only isolate faults). Therefore, the probability of fixing a system by performing a test is zero. Likewise, tests do not require any units from supply (UFS), so UFS, as a result of test performance, is also zero. To perform each analysis, the dominant action and best test cost analyses, the diagnostic module calculates UFS, time, and probability of success (POS) associated with the performance of the current activity under investigation and the next best activity. These calculations are then used to formulate the cost of each activity. The activity that exhibits the least cost is recommended.

PS_{SS}	= The probability of success of the next best activity (second step).
PS_{DA}	= The probability of success of the dominant action.
RRT_{DA}	= The time required to perform removal and replacement of the dominant action.
BTT	= The time required to perform the best test.
NAT	= The time required to perform the next best activity (second step).
Time	= The time to complete an activity normalized by its probability of success.
UFS	= The units from supply used to perform the activity.
POS	= The probability of success by performing the current activity and the next best activity (second step).
PTO_i	= The probability of ith test outcome i.
$PS_{SS}TO_i$	= The probability of success of the next best activity (Second Step) based on test outcome i.
n	= The number of test outcomes, which for a binary test would be 2 and for an MOT many.

a. Dominant Action Cost:

$$UFS = 1 + (1 - PS_{DA}) \tag{13}$$

Where $(1\text{-}PS_{DA})$ is equal to 0 if the next best activity is a test (no UFS for a test).

$$Time = RRT_{DA} + [(1 - PS_{DA}) \times (RRT_{DA} + NAT_{SS})] \tag{14}$$

$$POS = PS_{DA} + PS_{SS} \tag{15}$$

Where PS_{SS} is equal to 0 if the next best activity is a test (tests cannot fix).

$$\text{Rectification Cost} = \frac{(UFS \times Time)}{POS} \tag{16}$$

b. Best Test Cost:

$$UFS = \sum_{i=1}^{n} (1 \text{ unit} \times PS_{SS}TO_i) \tag{17}$$

Where $PS_{SS}TO_i$ is equal to 0, if the next best activity is a test (no UFS for a test).

$$Time = [\sum_{i=1}^{n} (NAT_{SS} \times PTO_i)] + BTT \tag{18}$$

$$POS = \sum_{i=1}^{n} (PS_{SS}TO_i \times PTO_i) \tag{19}$$

Where $PS_{SS}TO_i$ is equal to 0, if the next best activity is a test (test cannot fix).

$$\text{Test Cost} = \frac{(UFS \times Time)}{POS} \tag{20}$$

This idea expanded to the general case proved to be far more valuable than choosing a strategy based solely on time. Furthermore, the idea of cost could be readily expanded if the necessary data to clearly allocate costs associated with procurement, storage, transportation, maintenance, and test equipment for competing LRUs in the algorithm were obtained. Lacking such sophisticated data, a simple supply parts count can be an effective measure of the cost of a test versus an action decision.

Criticality

The IMIS diagnostic module determines whether there are critical faults in the plausible set. Each function has a designated list of potential faults which would render it inoperative. If a function is designated critical, then all potential faults associated with that function are designated critical. The plausible faults are compared to the designated critical faults. If there are any matches, the plausible faults are deemed critical, and the number of critical faults in the system are summed. The IMIS diagnostic module then examines each test and rectification by counting the number of critical faults spanned by that test or rectification and comparing the spanned fault count with the number of critical faults in the system under investigation. If the number of critical faults equals the spanned faults, the test or rectification is designated as critical. If more than one test is designated critical, the critical tests are ranked by the best test algorithm and the one with the greatest value is recommended first.

It is possible that all the critical faults in the plausible set can be corrected by a dominant action. All faults in the plausible set are evaluated for dominant action. If a dominant action has

been designated critical, the action is recommended. Otherwise, the best critical test is recommended. If no tests or rectifications are designated critical, diagnostics are continued until all critical faults are either exculpated or rectified.

Interleaving Actions

The IMIS diagnostic module algorithm, which evaluates tests and actions, is also used to generate a ranked list of options. The best test and dominant action loops already performed the basic evaluating and ranking functions, and, with minor modification, they were broadened to include the interleaving actions facility.

The first step after entering the loop is to initiate the multiple fault algorithms to generate a ranked list of fault sets, which represent the rectifications against which tests will be ranked. Next, using the methodology outlined for selecting a best test, the diagnostic module performs analyses and selects best tests for the given information and ranks these tests in decreasing order. The dominant action equation computes times to accomplish the rectification versus the best test and provides a decision whether to test or replace with the dominant action. The second step look-ahead analysis is then performed if any of the ranked actions are dominant and criticality is not invoked. If criticality is invoked, the dominant action is chosen. The test or action chosen becomes the first option in the list of interleaved actions. It is then removed from further comparison. The tests or actions not chosen are evaluated for the next interleaved option. This process of comparison using the best test, dominant action, and second step look-ahead analyses continues until the list of interleaved tests and actions has five entries, at which time the routine is terminated. For example, Action B is selected the first time as dominant action, and is chosen over Test 1, then Action B goes to the top of the list. It is removed from consideration and the comparison is executed again. If Test 1 dominates over all actions the second time, then it is placed on the list below Action B. It is then removed from consideration, and the loop is executed again. The third time, Test 2 is compared to all actions. Whichever activity dominates will be placed below the last activity placed on the list. A possible display of the top five options would be in the following format:

1. ACTION B
2. TEST 1
3. ACTION C
4. ACTION A
5. TEST 2

Reinitialization/Change in Symptom

The IMIS diagnostic module can react to changes in the diagnostic situation by updating parameters during diagnostics. Changes in symptoms might occur if a symptom is discovered or removed during rectification. As the diagnostic module is executed and as the technician applies the information to the problem, certain information is gained. Tests are passed/failed, and possible faults are exculpated from the plausible set. This information is useful to the diagnostic module because it reduces the problem's complexity and brings the solution closer.

For example, assume the IMIS diagnostic module begins diagnostics with a set of symptoms implicating a given number of faults. Symptoms are eliminated as faults are isolated and components rectified. Assume a specific symptom has been eliminated and, with it, several faults are removed from consideration. There still remain other symptoms and faults to be removed; however, the problem's complexity may be reduced. One of the exculpated faults might be implicated by one of the remaining symptoms. By knowing that this fault is exculpated, the

plausible set of faults for the symptom still being investigated is reduced, and the resulting computations are simpler and quicker. The ability to account for a change in symptoms is important if the diagnostic module is to effectively attack a problem.

Whenever a rectification or maintenance action is completed, the module performs a system check and returns symptoms. Any changes in the state of the fault/symptom matrix are updated by user inputs and the diagnostic module simply adds or deletes information as necessary. Information is not lost, and any changes in the state of the problem are handled and incorporated in the succeeding diagnostic steps. Data are input throughout the process; the loop is never exited. User input menus to identify symptom changes within the diagnostic loop provide the diagnostic module with a recursive network that is reinitialized at the start of each diagnostic sequence iteration.

IV. IMIS DIAGNOSTIC AND USER INTERFACE FUNCTIONS EMPLOYED DURING A FIELD DEMONSTRATION ON THE F-16 A/B RADAR SYSTEM

The IMIS diagnostic module was combined with an automatic technical order data presentation system and custom designed user interface for a field test/demonstration on the F-16 A/B radar system. The software was loaded on a portable computer maintenance aid developed by the AFHRL. The demonstration/field test was performed at Homestead Air Force Base, Florida.

The user/computer interface functions developed for and demonstrated at Homestead AFB provided a good user interface designed for ease of use and flexibility. Capabilities demonstrated included manual and automatic downloading of BIT results from the aircraft MIL-STD-1553 data bus, automatic collection of diagnostic and repair data in a log file, a suspend function, a back up function, the ability to display several ranked lists of available diagnostic actions, and the ability to select actions from a Table of Contents list.

Manual/Automatic Symptom Loading

A hardware/software controller to activate and use the aircraft data bus was incorporated on the portable computer. Two forms of symptom loading (functional check result entries) were input back to the IMIS diagnostic module for initialization: automatic and manual. Figure 8 represents the flow of information for automatic and manual feedback in IMIS diagnostics.

During initialization, the diagnostic module provides the maintenance technician with the opportunity to enter the fault/symptom information manually. Manual fault/symptom information was provided to the maintenance technician from pilot input data and previously performed MIL-STD-1553 data bus downloads. These were presented on the debrief Air Force Technical Order (AFTO) Form 349 presented at the start of the demonstration.

Upon completion of manual fault/symptom entry, automatic feedback through the MIL-STD-1553 data bus initialized BITs and returned symptoms which were directly input to the IMIS diagnostic module for symptom verification and entry. For example, the Fire Control Computer (FCC), through the MIL-STD-1553 data bus, initialized system and fault checks, and received and stored information from BITs. The portable computer maintenance aid initialized system fault checks and downloaded system and fault information directly from the MIL-STD-1553 data bus. Figure 9 illustrates the MIL-STD-1553 data bus with some system connections and provides an illustration of its usefulness.

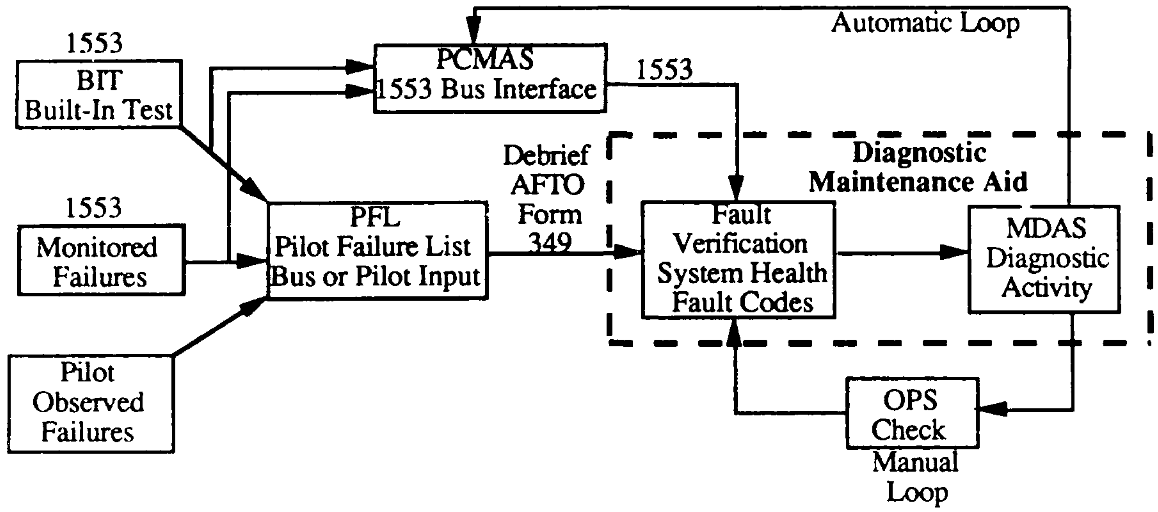


Figure 8. Flow of Diagnostic Information.

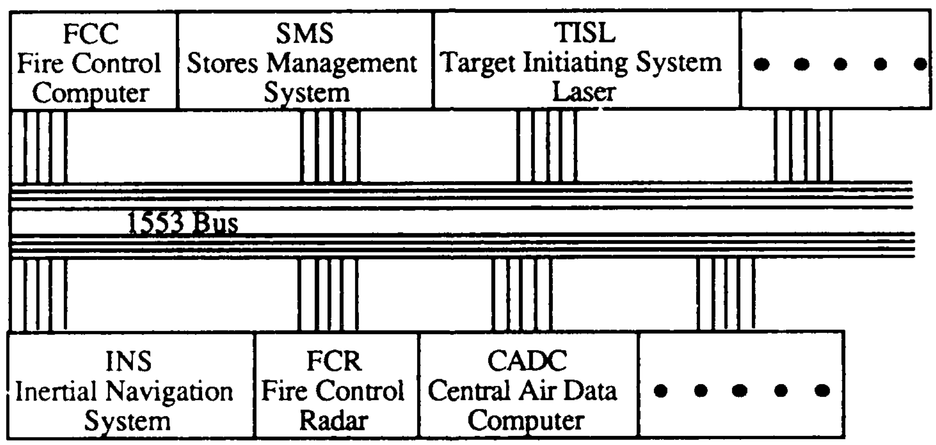


Figure 9. MIL-STD-1553 Data Bus Schematic.

Automatic Data Collection

Today's maintenance data collection systems fall short of providing the detailed data needed to make the diagnostic aiding system really efficient. For example, no data are collected on test times, fault occurrence rates, access and closure times, and remove and replace times. Data that are collected, such as component failure rates and total task times, are accumulated by aircraft type which tends to hide individual location variations. For example, an aircraft in a coastal environment may have corrosion problems with a particular component, while an aircraft of the same type at a landlocked base may not. The diagnostic aiding system has been developed with the ability to collect, collate, and update all of these pertinent parameters on a real-time basis. However, the facility to capture and store this information over an extended period of time for later use does not yet exist. When implemented for analysis, this capability will provide the technician with diagnostic data tailored to local peculiarities of environment and operations.

Log File

A utility to record major actions taken in a diagnostic sequence is implemented in the IMIS diagnostic module. This utility is called the log file. The log file has numerous applications not only in the IMIS diagnostic module but also in the general maintenance area. The ability to record a diagnostic sequence facilitates the re-creation of that diagnostic sequence at a later date for either review or training purposes. The diagnostic sequence in a particular log file may be examined side by side with other sequences in order to compare diagnostic paths. Furthermore, diagnostic sequences may be extracted from the log file to facilitate training activities as examples or exercises for students, and can also be analyzed for information concerning the supplies, equipment, manpower, and costs associated with specific diagnostic sequences, equipment, or operating locations.

The log file allows the IMIS diagnostic module to be a more complete diagnostic tool by providing information about actions taken during a repair session. This information is intended to be analyzed out of the IMIS diagnostic module. It has already shown its utility in the development of a feedback analysis tool which generates manpower, spares, support equipment, and other items of logistics concern.

The operation of the log file is fairly simple and straightforward. The log file is implemented using a major keystroke accumulator which saves the actions taken and the time required to complete those actions. At the end of a diagnostic session, the information is written to an external file that can be accessed outside the IMIS diagnostic module.

Suspend Function

This function was developed to deal with instances in which the diagnostic sequence needs to be temporarily halted (unavailability of parts or equipment, time constraints, etc.). The suspend function allows the technician to permanently save the sequences and actions for use when diagnostics are resumed. This saved information includes all aspects of the diagnostic evaluation including the complete path taken to the point at which work was suspended. In a typical situation, the technician performs fault diagnosis using the IMIS diagnostic module until reaching a point at which the diagnostics could no longer continue. At this point, the suspend function is executed saving all work. When ready to continue diagnostics, the technician simply loads the suspended file and returns to the exact location exited.

Back Up Function

A complete diagnostic aid should give the user as much pertinent information and flexibility in operation as possible. A technician progressing through a diagnostic sequence may wish to back up to a designated test or rectification previously performed, or may wish to review steps already completed within a technical order sequence. The back up function has been researched and is incorporated into the IMIS diagnostic module. This function allows the user to back up to any action previously performed in the diagnostic session or to a particular step within a technical order sequence. Upon performing a back up, all subsequent diagnostic information acquired from actions or sequences of technical orders performed is deleted. This function is useful if there is a change in the ability to perform an action or an action was performed improperly.

Display Tests/Rectifications

During a diagnostic sequence, the technician may wish to view any options that are available to expedite or complete diagnostics. This is an important part of the man-machine interface of the diagnostic tool. Technicians must be able to incorporate their maintenance expertise

in any given diagnostic sequence. A diagnostic aid which ignores operator expertise is not only inflexible but also impractical. In many cases, the technician progressing through diagnostics is able to come to conclusions about the problem due to sheer intuition or similar past experiences.

This concept was demonstrated by lists of all available rectifications and tests available to the technician. The feature enabled the technicians to evaluate all the possible options available to isolate or rectify a problem. Upon viewing these lists, the technician could evaluate the situation and either comply with the machine's recommendations or use personal expertise and experience to select a different option which would isolate or rectify the problem. Additionally, such lists were helpful in evaluating "what if" questions and enhancing the training capabilities of the tool.

Display Interleaved Tests/Rectifications

Studies completed during research of human interface issues revealed that technicians desire access to as much information as possible about the diagnostic problem on which they are working. Therefore, the need for ranked isolation and repair options resulted in the implementation of a display interleaved actions list. This enhancement to the diagnostic module data display was achieved by the interleaving of actions analysis described in Section III. The technician selected the menu function of interleaved actions, and a mixed hierarchical list of the top five actions provided convenient viewing of the options that will best lead to fault rectification. The maintenance technician then had the opportunity to either select the recommended option or choose from among alternative options presented in the list.

Review Previous Actions

Another function which enhanced flexibility of operation and gave the user more information is the "review previous actions" function. This facility was implemented to allow the technician to view all the tests and actions already accomplished in the diagnostic sequence. This feature is accessed via a function key. When called, it displayed to the user a complete ordered list of all tests and actions accomplished during the diagnostic session, along with the result of that activity. This type of function gave the technician information as to what was accomplished which made for a more efficient diagnostic sequence by avoiding repeated actions. This feature is very useful when a technician must complete a diagnostic activity initiated by another or when the diagnostic activity has been suspended.

Table of Contents

In any interaction with technical order data, eventually the user will want to choose a new point of entry into the data. Consequently, a table of contents facility was created to give the users an interface with a "look and feel" much like they were familiar with in using AFTOs. The feature provided a bonus capability in diagnostics as well. In some cases, especially with immature data bases, there may arise occasions when the diagnostic module is simply unable to provide further assistance in fault isolation. In such a case, it is essential that all the technical data available to describe a system and prescribe repair actions be available to the technicians. At that point, they will be working on the basis of intuition and their own knowledge of the system, and it is imperative they have access to all available data. Hence, a table of contents facility was provided on the portable computer.

V. CONCLUSION

The IMIS diagnostic module is the implementation of a powerful diagnostic strategy capable of handling multiple faults, MOTs, critical functions, and equipment availability. The strategy is founded in a fault-based approach which overcomes limitations of a component connection analysis yet avoids the needless detail of low-level, bit-and-piece analyses.

The development of fault/component modeling techniques provides a flexible reachability matrix which computer evaluations could attack. This reachability matrix maps rectifications (components), tests, faults, and symptoms, providing the relationship needed for analysis, and creating a structure for repair of a faulty system. The reachability matrix also allows for the incorporation of fault probabilities and, when expanded to include more than one symptom, demonstrated that the cause of a faulty system could be two or more independent faults which could be resolved by multiple fault evaluation.

Upon development of the reachability matrix, the theory of integrating fault isolation and rectification strategies provides the most direct route to fixing an aircraft with the least amount of time expended. A fault isolation strategy alone limits the steps taken to isolate a faulty component, but rectification of that component still needs to be performed. Including rectifications in the diagnostic analyses allows a technician to rectify an aircraft system in the least amount of time by recommending actions that are so likely to solve the problem at hand that they are recommended prior to testing.

Other considerations had to be theorized and integrated into the diagnostic aid: at what cost and information gain would the performance of one selected activity out rank the performance of another, and what are the next step ramifications of each activity. This development involved the incorporation of time, unit cost, probability of occurrence, information gained, and forecasting of second step events.

ACRONYMS

AFHRL	-	Air Force Human Resources Laboratory
AFTO	-	Air Force Technical Order
BIT	-	Built-In Test
F	-	Fault
FCC	-	Fire Control Computer
FCK	-	Functional Check
FR	-	Failure Rate
IMIS	-	Integrated Maintenance Information System
LRU	-	Line Replacement Unit
MOT	-	Multiple Outcome Test
MTBF	-	Mean Time Between Failures
$P(F_n)$	-	Probability of Fault n Occurring
POS	-	Probability of Success
PS	-	Plausible Set
R	-	Rectification
R&D	-	Research and Development
RIU	-	Remote Interface Unit
RRT	-	Removal and Replacement Time
RT	-	Receiver Transmitter
S	-	Symptom
SRU	-	Shop Replacement Unit
T	-	Test
TT	-	Test Time
UFS	-	Units From Supply

GLOSSARY

Action. A diagnostic or corrective procedure performed by a maintenance technician.

Availability. A component's or test equipment's obtainability for use in the diagnostics process.

Best Rectification. A diagnostic software algorithm that chooses the optimum from among available rectification actions.

Best Test. A diagnostic software algorithm that chooses the optimum test from among those available at any point in the diagnostic sequence.

Component. The lowest physical level of indenture on which a maintenance technician at a given level of maintenance (i.e., organizational, intermediate, and depot (O, I, or D)) will normally work. For example, an organizational level maintenance technician would consider a Line Replacement Unit (LRU) as a component; whereas, an intermediate level technician would consider the LRU an end item and the Shop Replacement Unit (SRU) a component.

Criticality. A measure of need for a particular system capability. For example, a fault in an air-to-ground function might not be critical for an air defense sortie, whereas a fault in an air-to-air function would be critical for the same sortie requirement.

Dominant Action. A rectification action whose likelihood of success is so great that it is recommended before available tests that would reduce the plausible set.

Fault. The manifestation, through either inference or direct observation, of a failure within a system.

Functional Check. A test performed to ensure that a rectification action has been successful in restoring a system to operational status.

Mean Time Between Failures (MTBF). The unit of reliability used in this program as a predictor of fault likelihood. Its inverse is the failure rate.

Multiple Faults. An event where two or more faults exist at the same time in a given system.

Multiple Outcome Test (MOT). A test procedure without a binary pass/fail result. The procedure may have any number of outcomes; however, each outcome is unique and distinguishable from all other outcomes

Plausible Set. The set of possible faults that could logically have led to an observed or indicated faulty condition. The elements in this set of faults contain single faults or combinations of faults that are not redundant.

Rectification. The repair of a fault(s) which may alleviate a symptom or set of symptoms.

Repair Time. The time required to complete system repair after a fault is isolated. It may include access times. It will include reinstallation of original components removed unnecessarily as part of diagnostics, secure and closure, and final functional check.

Symptom. A machine-generated code or verbal description indicating a malfunction exists (e.g., "Receiver, no audio").

Test. A prescribed sequence of actions whose result will implicate or exonerate a set of faults.

Test Time. The time required to perform a test. It includes access time, time to gather necessary test equipment and tools, time to conduct the test procedures, and time needed to record/interpret test results.